**SQL Injection Manual**

**Descripción**

SQL Injection es una vulnerabilidad que ocurre cuando una aplicación web permite a los usuarios enviar datos que se insertan en una consulta SQL sin ser correctamente filtrados o escapados. Esto permite a un atacante manipular las consultas SQL que la aplicación ejecuta, lo que puede resultar en el acceso no autorizado a los datos, la modificación de datos, o incluso la toma de control del servidor.

**Ejemplo de Vulnerabilidad**

Considere una aplicación web que utiliza una consulta SQL para autenticar a un usuario:

php

Copy code

<?php

$username = $\_POST['username'];

$password = $\_POST['password'];

$query = "SELECT \* FROM users WHERE username = '$username' AND password = '$password'";

$result = mysqli\_query($conn, $query);

if (mysqli\_num\_rows($result) > 0) {

// Usuario autenticado con éxito

echo "Welcome, " . $username;

} else {

// Fallo en la autenticación

echo "Invalid credentials";

}

?>

Si un atacante proporciona el siguiente username y password:

vbnet

Copy code

username: admin'--

password: anything

La consulta SQL se convierte en:

sql

Copy code

SELECT \* FROM users WHERE username = 'admin'--' AND password = 'anything'

El comentario -- hace que todo después de admin se ignore, autenticando al atacante como el usuario admin sin necesidad de conocer la contraseña.

**Ejemplo de Explotación**

1. **Bypass de autenticación:**
   * **Input del atacante:**

bash

Copy code

username: ' OR '1'='1

password: ' OR '1'='1

* + **Consulta resultante:**

sql

Copy code

SELECT \* FROM users WHERE username = '' OR '1'='1' AND password = '' OR '1'='1'

* + **Resultado:** La consulta siempre será verdadera, permitiendo el acceso sin autenticación.

1. **Extracción de datos:**
   * **Input del atacante:**

vbnet

Copy code

username: ' UNION SELECT null, table\_name FROM information\_schema.tables --

password: anything

* + **Consulta resultante:**

sql

Copy code

SELECT \* FROM users WHERE username = '' UNION SELECT null, table\_name FROM information\_schema.tables --' AND password = 'anything'

* + **Resultado:** Los nombres de las tablas de la base de datos se devolverán como resultado de la consulta.

**Mitigación**

1. **Usar consultas preparadas (prepared statements):**

Las consultas preparadas son una manera segura de ejecutar consultas SQL en lugar de concatenar directamente las entradas del usuario en la consulta SQL.

* + **PHP con MySQLi:**

php

Copy code

$stmt = $conn->prepare("SELECT \* FROM users WHERE username = ? AND password = ?");

$stmt->bind\_param("ss", $username, $password);

$stmt->execute();

$result = $stmt->get\_result();

if ($result->num\_rows > 0) {

echo "Welcome, " . $username;

} else {

echo "Invalid credentials";

}

* + **Python con SQLite:**

python

Copy code

import sqlite3

conn = sqlite3.connect('example.db')

cursor = conn.cursor()

cursor.execute("SELECT \* FROM users WHERE username = ? AND password = ?", (username, password))

if cursor.fetchone():

print("Welcome, " + username)

else:

print("Invalid credentials")

1. **Sanitizar y validar entradas:**

Asegúrese de que las entradas del usuario se validen y saniticen adecuadamente para evitar la inyección de SQL. Use funciones como mysqli\_real\_escape\_string() en PHP para escapar caracteres especiales en una cadena para usar en una consulta SQL.

1. **Principio de privilegios mínimos:**

Asegúrese de que la cuenta de la base de datos utilizada por la aplicación tenga los privilegios mínimos necesarios. No use cuentas de base de datos con privilegios administrativos para las operaciones de la aplicación.

**Herramientas de Prueba**

* **SQLMap:** Una herramienta de código abierto que automatiza la detección y explotación de fallos de inyección SQL.
* **Burp Suite:** Un proxy de seguridad web que incluye herramientas para detectar y explotar inyecciones SQL.

**Resumen**

La inyección SQL es una vulnerabilidad crítica que puede comprometer la seguridad de una aplicación web. El uso de consultas preparadas, la sanitización de entradas y la implementación del principio de privilegios mínimos son esenciales para proteger contra estos ataques.